

[Log in](#)[Start for free](#)[IBM Cloud Learn Hub](#) > [DevOps](#)

DevOps

[DevOps](#)[Site feedback](#)

2 October 2019

By: IBM Cloud Education



In this guide, learn about DevOps, its processes, and how it's an evolution of existing development models.

Table of Contents

[What is DevOps?](#)[Benefits of DevOps](#)[DevOps tools](#)[DevOps methodologies, principles, and strategies](#)[Continuous integration](#)[The DevOps lifecycle](#)[DevOps vs. ...](#)[DevSecOps](#)[The DevOps toolchain](#)[Let's talk](#)

What is DevOps?

With modern businesses moving at the speed of cloud, DevOps has become an increasingly common approach to software delivery that development and operations teams use to build, test, deploy, and monitor applications with speed, quality, and control.

DevOps is essential for any business aspiring to be lean, agile, and capable of responding rapidly changing marketplace demands. It is an approach on the journey to lean and agile software delivery that promotes closer collaboration between lines of business, development, and IT operations while removing barriers between your stakeholders, and your customers.

Site feedback

To be essential to customers, all stakeholders in the delivery process need to collaborate. Development teams need to design, develop, deliver and run the software as quickly and reliably as possible. Operations teams need to identify and resolve problems as soon as possible by monitoring, predicting failure, managing the environment and fixing issues. Combining this common approach across Dev and Ops with the ability to monitor and analyze bottlenecks and optimize as quickly as possible gives you DevOps—a collaborative approach across business, development, and operation stakeholders to deliver and run reliable software as soon as possible.

Andrea Crawford, Distinguished Engineer and CTO of DevOps, gives a deeper look into the evolution of DevOps and its underlying goals in her blog post, "[What is DevOps?](#)" and the following video:

[Watch the video](#)

Explore more about DevOps tools and approaches with [IBM DevOps](#).

Benefits of DevOps

Implementing a DevOps practice can add value to your organization through a number of benefits, including the following:

- Faster code delivery
- Faster time to market

Let's talk

- Higher-quality software
- Improved collaboration between developers and operations
- Decreased time to resolution for fixing bugs and vulnerabilities
- A culture that brings business, development, and operations together for improved responsiveness to market demands

DevOps tools

DevOps tools cover a range of processes within the software development life cycle:

- **Define and plan**, which focuses on planning DevOps workflows for iterations, release management, and issue tracking. Notable tools or tool vendors in this space include Atlassian, CA Technologies, IBM, iRise, and Jama Software.
- **Code, build, and configure**, which focuses on code development and review, source code management, and code merging. Notable tools/tool vendors include BitBucket, Electric Cloud, GitLab, GitHub, and IBM.
- **Test**, which verifies that the quality of the software release and code are maintained throughout the development process and that the highest quality deploys to production. Notable tools/tool vendors include Delphix, FlawCheck, HP, IBM, Microsoft, Parasoft, SonarSource, Skytap, and ThoughtWorks.
- **Packaging and preproduction**, which refers to the activities involved once the release is ready for deployment; it's also called staging or preproduction. Notable tools/tool vendors include IBM, Inedo's ProGet, Jfrog's Artifactory, Sonatype Nexus repository.
- **Release, deploy, and orchestration**, which is the process of actually releasing software and usually involves change management, release approvals, release automation, schedule orchestration, provisioning, and deploying into production. Tools/tool vendors in this space include Automatic, Clarive, BMC, IBM, Flexagon, VMware, and XebiaLabs.
- **Continuous management and configuration** includes continuous configuration automation, configuration management, and infrastructure as code. Notable tools/tool vendors include Ansible, Chef, IBM, Puppet Labs, Otter, and Salt.
- **Monitoring** reports application performance and helps identify issues impacting user experience. Tools/tool vendors include Big Panda, IBM, New Relic, Plumb, and Wireshark.

For a closer look, watch our video, "Introduction to DevOps tools":



Introduction to DevOps tools

DevOps methodologies, principles, and strategies

DevOps methodologies

DevOps grew out of Agile. Agile is a way of producing software in short iterations on a continuous delivery schedule of new features and bug fixes in rapid cycles from two to four weeks. In contrast, DevOps brings the development and operations teams together to focus on eliminating silos to decrease time of addressing customer feedback and break down bottlenecks to enable continuous software delivery. Consequently, they can build, test, and release software more quickly with as much efficiency and speed as possible.

Not only does DevOps involve more of the organization in the development process— Let's talk lines of business, suppliers involved in software delivery, and consumers themselves—but it does it

in a way that speeds development and improves quality, according to a [white paper on software-driven innovation](#). This can lead to the creation of a culture of innovation when you adopt DevOps methodologies, allowing you to collaborate and react with agility to changes in the market.

DevOps methodologies include the following:

- **Continuous integration**, which is where coding, building, integrating, and testing take place.
- **Continuous delivery**, which includes continuous integration, but mainly focuses on product releases.
- **Continuous deployment**, which focuses on automating releases of projects as soon as possible.
- **Operate** for conducting the development operations of configuration management and continuous monitoring.

Site feedback

For a deeper dive into some of these DevOps methodologies, see our "[DevOps 101 \(5:58\)](#)" series of videos.

DevOps principles

At the heart of [DevOps principles](#), you will find the idea of collaborative learning and collaborative relationships between development and operations. They focus on increasing the pace of planned work for higher deployment rates, while also upgrading the reliability, stability, resilience, and security of the production environment. To establish an organization based on DevOps principles, you need to emphasize this holistic, whole-system approach across not just the development and operations departments but also every surrounding department and support organization within the company. In return, your whole system should be used to shape your organizational goals.

DevOps principles include:

- Process improvement initiatives to truncate feedback loops to continuously implement needed bug fixes and vulnerability remediation earlier and more cost effectively
- Continual experimentation that encourages risk-taking and learning from success and failure, so continuous attempts will lead to future success and mastery
- Learners becoming teachers and passing along their acquired knowledge to their peers

Let's talk

customer feedback, and small, rapid releases. While Agile applies to a single practice, DevOps applies to the entire lifecycle of a project or application. And while you shouldn't necessarily have to choose between an Agile and DevOps approach, you should know how they differ in their application.

Agile enables developers to deliver their functions every two weeks to respond to changing business needs. DevOps focuses on the operational side of the software development lifecycle, reducing handoffs from developer to operations teams, reducing the time to testing and deploying code, and decreasing errors and downtime of operational systems.

[Site feedback](#)

DevOps vs. SRE

Site Reliability Engineering (SRE) is an approach to cloud operations that ensures that continuously delivered, cloud-based applications run efficiently and reliably by using software engineering and automation solutions. The key concept is engineering, which includes a data-driven approach to operations, a culture of automation to drive efficiency and reduce risk, and hypothesis-driven methodology in incident, performance, and capacity tasks. And while SRE didn't evolve from DevOps, it is in alignment. In addition, SRE is an [engineering cloud approach to operations](#) that applies modern cloud design patterns to code, lasting solutions to service issues, focusing at the application level and using automation to manage the infrastructure layer. SRE will involve some or all of the following tasks:

- Eliminating performance bottlenecks by refactoring services into more scalable units
- Isolating failures through the use of the [cloud-native](#) design patterns
- Creating runbooks to ensure fast service recovery
- Automating day-to-day ops processes

What's most exciting is that SRE is the practice of [using DevOps to protect your infrastructure](#). This leads to the role of the Site Reliability Engineer becoming a hybrid DevOps role—part developer, part system administrator. Because SREs have feet in both worlds, they should be able to shift smoothly from fixing code issues to solving operational issues. This flexibility allows quick redeployment of SRE resources as business priorities change and products mature.

Let's talk

DevOps vs. SysOps

- Using DevOps automation to improve efficiency
- Giving continuous feedback to the entire organization
- Incentivizing development, test, and deployment teams to collaborate on shared goals

DevOps strategies

Continuous integration, continuous delivery, continuous deployment—by now, you’ve got the idea that DevOps builds on the concept of continuity. That’s not by accident; DevOps strategy focuses on the enterprise capability for continuous software delivery that enables customers to seize market opportunities and reduce time to customer feedback. As a strategy, it involves people and is not focused on any one technology or practice. It enables people to collaborate across teams and work on problems anywhere in the software development life cycle. It always has the customer as its sole focus, and the customer is everyone’s responsibility. With a DevOps strategy you can do the following:

Site feedback

- Accelerate the delivery of reliable software
- Balance speed, cost, quality, and risk with increased capacity to innovate
- Reduce time to customer feedback with improved customer experience

Continuous integration

Within the rapidly evolving field of product and application lifecycles, continuous integration maintains a simple objective—to frequently code, build, integrate, and test the work of all developers on a software project at least once a day. As a cornerstone of DevOps, [continuous integration](#) keeps the code of an individual developer from drifting too far afield from the work of the development group. This technique continually merges source code updates from all developers on a team into a continuous integration server.

If a failure occurs, the development team can refocus and fix it before making any code changes. While this may seem disruptive, in practice it focuses the development team on a single stability metric: a working automated build of the software, according to the [Carnegie Mellon University Software Engineering Institute](#).

In the following video, Eric Minick takes a closer look at continuous integration:

Let’s talk

What is Continuous Integration?

[Site feedback](#)

The DevOps lifecycle

As the speed of business and [cloud computing](#) continues to increase, the DevOps lifecycle has also accelerated to support them. DevOps offers an approach that enables business, development, and operations to continuously collaborate to deliver software to incorporate customer feedback in less time and take advantage of gaps in the market where there is not currently a solution.

Following a lean software delivery process, the [DevOps lifecycle](#) will allow your team to identify waste, duplication of efforts, and bottlenecks in the process by establishing a continuous innovation and improvement feedback loop among customers, lines of business product managers, software and product development, and operational manufacturing and support. It will reduce time to obtain and act on customer feedback, accelerate software delivery, and balance speed, quality, cost, and risk.

DevOps vs. ...

DevOps vs. Agile

As stated previously, DevOps is an evolution of Agile. The Agile development model prescribes short, concentrated efforts on individual product elements, typically completed in two [Let's talk](#) weeks. Overall, the iterative approach of Agile has the same objectives as DevOps: collaboration,

Up to this point, we have made it clear that DevOps combines the functions of development and operations. An older term, SysOps (system operations), also known as systems administration, sometimes seems to confuse people and the two become conflated. So to explain the [difference between DevOps and system administration](#), InfoWorld states that DevOps focuses on the higher-level processes across the company and has end-to-end responsibility for products, while SysOps is devoted to configuring and maintaining individual computer systems and smaller scope projects.

But SysOps does have a role to play in the software development life cycle. For example, a developer might ask SysOps to stop and start a test CICS or IMS region, or managers might do the same for production systems in the situation when a production system requires this kind of intervention with a [focus on ops](#), according to Joseph Gulla, adjunct professor of computing at Nova Southeastern University. SysOps supports a consistent manner to do deployment and change of an application to help out the developer team.

[Site feedback](#)

DevSecOps

In the past, some people thought that DevOps and security mixed like oil and water—meaning they didn't. But according to [Wired](#), DevOps benefits security, and with the right automation and operational tools, you can inject security earlier into the development process and increase the security of production code.

This combination of DevOps and security gives rise to secure DevOps, or DevSecOps. To [implement secure DevOps](#), you must consider five facets of security:

- **Secure engineering:** Products are developed with strong security and comply with appropriate security standards.
- **Secure deployment and operations:** Cloud platform and applications are configured and deployed securely, tested for security vulnerabilities, patched, and have their bugs fixed.
- **Separation of duties:** Users access what's required for their job duties (i.e., the principle of least privilege).
- **Availability and business continuity management:** For 99.999 percent availability of infrastructure, runtime components, and management components.

Let's talk

Security evaluation and learning: You maintain security functions and properties in code and

[IBM UrbanCode Deploy](#) is used to automate the deployment of software to different environments. It mitigates the risk of managing multiple application configurations, integrations, and versions across many ecosystems.

[IBM UrbanCode Velocity](#) is used to orchestrate the overall delivery of the entire software solution. It provides insight into how value flows through each delivery pipeline and highlights where bottlenecks are disrupting the flow.

[Site feedback](#)

DevOps and IBM Cloud

DevOps requires collaboration across business, development, and operation stakeholders to expeditiously deliver and run reliable software. Organizations that use DevOps tools and practices while transforming their culture build a powerful foundation for [modernizing their applications](#) on their [journey to cloud](#).

Take the next step:

- Explore IBM tools to support a DevOps approach, including [IBM Architecture Room Live](#), [IBM Rational Test Workbench](#), [IBM UrbanCode Deploy](#), and [IBM UrbanCode Velocity](#).
- See how [IBM DevOps](#) can help you shorten releases, improve reliability and stay ahead of the competition.
- Check out the eBook, [Putting the Ops into DevOps for Dummies](#) (PDF, 2.1MB).

Read what analysts say:

- [Gartner Magic Quadrant for Application Release Orchestration](#), Q3 2018
- The Forrester Wave report: [Continuous Delivery and Release Automation](#), Q4 2018
- [IDC MarketScape: Worldwide Cloud Testing, ASQ SaaS 2017 – 2018](#)

Get started with an [IBM Cloud account today](#).

Follow IBM Cloud

Let's talk

- **Security evaluation and learning.** You maintain security functions and properties in code and services as threats evolve and you discover new vulnerabilities.

Check out the video "What is DevSecOps?" to learn more:



Site feedback

The DevOps toolchain

Both culture and tools will help you realize the DevOps methodology. In achieving this goal, it is vital to ensure data sharing and collaboration can occur across the delivery tools pipeline. Incumbent and open source tools—especially where investment has been made and still needs to be adopted—can be used as part of the process in conjunction with IBM tools that address the critical challenges faced by DevOps teams:

[IBM Architecture Room Live](#) is used to design and communicate new architectures. It is a browser-based design and architecture tool allowing real-time, multi-user, multi-site collaboration via a virtual architecture “whiteboard.”

[IBM Rational Test Workbench](#) is used to virtualize the test environment dependencies that are not available when the team needs to start testing. It enables the creation, execution, and reporting of the quality of the integration, functional and performance capabilities as well as the end to end business and technical scenarios.

Let's talk

IBM Cloud News connects you to insight and information you can put to work right away—straight from the minds of IBM Cloud experts, IBM customers, and business and IT leaders.

Email subscribe
RSS

All topics

Site feedback

You May Also Be
Interested In



IBM Cloud Continuous
Delivery Tekton Pipelines
Tools and Resources

By: Eric Jodet

13 August 2020



The Kubernetes Effect:
Igniting Transformation in
Your Team

By: Briana Frank

12 August 2020



Toolcha
Tekton

By: Eric Joc

14 July 202

Be the first to hear about news, product updates, and innovation from
IBM Cloud

Let's talk

Get updates to your inbox. →

Learn

What is Cloud Computing

What is Kubernetes

What are Containers

What is Serverless Computing

What is DevOps?

Data centers

Products

Solutions

Customers

Security

Site feedback

Build

Get started

Docs

Courses

Practices

Architectures

Toolchains

Engage

Support

Services

Garage

Let's talk

Partners

Blog

Newsletter

Twitter

YouTube

Your account

Sign up

Log in

Get help

Login issues

Site feedback



Let's talk